Requested Patent: JP11212831A

Title:

LOG MANAGEMENT OF INTER-OBJECT COOPERATIVE OPERATION AND SYSTEM AND METHOD FOR EVENT TRACKING ;

Abstracted Patent: JP11212831 ;

Publication Date: 1999-08-06 ;

Inventor(s):

FUJIWARA TOOSHI; KUWADA TAKASHI; YAMAMOTO HIDEAKI; MASUKI RYOSUKE; UCHIKAWA ATSUSHI; ICHIKI NOBUHIKO ;

Applicant(s): NTT DATA CORP; SUMITOMO BANK LTD ;

Application Number: JP19980013675 19980127 ;

Priority Number(s): ;

IPC Classification: G06F11/34 ; G06F13/00 ; G06F15/16 ;

Equivalents:

ABSTRACT:

PROBLEM TO BE SOLVED: To facilitate log collection and event tracking regarding cooperative operation between plural objects distributed on a network. SOLUTION: Event trackers 260 are arranged on each node 220. Each of the event trackers 260 records logs of events that all the objects within a self node perform in a log file by the object. Each of the event logs includes write-in time, names of objects, transaction IDs, repeating point counters for indicating what the object's order is from a starting point of transaction and kind of events. The event trackers 260 also collect logs of all the events of a specific transaction by retrieving from a log file within its own node by depending upon a transaction ID and asking the event tracker of other nodes for retrieval. Then, the event trackers 260 resolve a progress of the transaction by a name of process, a repeating point counter and the event kinds of a collected event log.

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報（A）**　(11)特許出願公開番号

特開平11－212831

(43)公開日　平成11年(1999)８月６日

(51)Int.Cl.[6]		識別記号	FI		
Ｇ０６Ｆ	11/34		Ｇ０６Ｆ	11/34	Ａ
	13/00	３５５		13/00	３５５
	15/16	４３０		15/16	４３０Ｂ

審査請求　未請求　請求項の数26　ＯＬ　（全 21 頁）

(21)出願番号　特願平10－13675

(22)出願日　平成10年(1998)１月27日

(71)出願人　000102728
株式会社エヌ・ティ・ティ・データ
東京都江東区豊洲三丁目３番３号

(71)出願人　592038649
株式会社住友銀行
大阪府大阪市中央区北浜４丁目６番５号

(72)発明者　藤原　遠
東京都江東区豊洲三丁目３番３号　エヌ・ティ・ティ・データ通信株式会社内

(72)発明者　黄檗　隆
東京都江東区豊洲三丁目３番３号　エヌ・ティ・ティ・データ通信株式会社内

(74)代理人　弁理士　上村　輝之

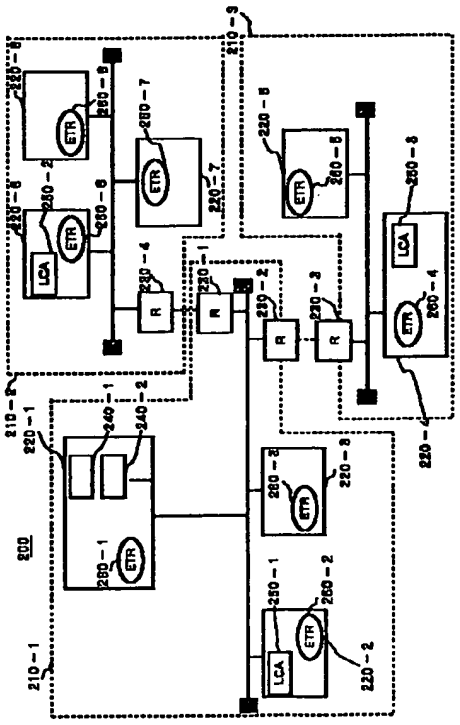
最終頁に続く

(54)【発明の名称】　オブジェクト間協調動作のログ管理およびイベント追跡のための方式および方法

(57)【要約】

【課題】　ネットワーク上に散在する複数のオブジェクト間の協調動作に関するログ回収やイベント追跡を容易にする。

【解決手段】　各ノード２２０にイベントトラッカ２６０が配置される。各イベントトラッカ２６０は、自ノード内の全てのオブジェクトの行ったイベントのログを、各オブジェクト別にログファイルに記録する。各イベントログには、書き込み時刻と、オブジェクト名と、トランザクションＩＤと、そのオブジェクトがトランザクションの起点から数えて何番目の中継点であるかを示す中継点カウンタと、イベント種別とが含まれている。イベントトラッカ２６０はまた、必要に応じ、トランザクションＩＤを頼りに自ノード内のログファイルから検索し且つ他のノードのイベントトラッカに検索を依頼することによって、特定のトランザクションの全イベントのログを収集する。そして、イベントトラッカ２６０は、収集したイベントログのプロセス名と中継点カウンタとイベント種別とにより、そのトランザクションの経過を解明する。



【特許請求の範囲】
【請求項1】　複数のノードを有する通信ネットワークと、
前記複数のノードに分散して配置された複数のオブジェクトと、
各ノードに配置されたログファイルと、
前記各ノードに配置され、前記各ノード内の前記オブジェクトが実行したイベントのイベントログを前記ログファイルに記録するログライタとを備え、
前記複数のオブジェクトは、相互間で協調動作を依頼しながら1つのトランザクションを連携して処理することができ、
各オブジェクトは、他のオブジェクトに協調動作を依頼するとき、前記トランザクションを識別するためのトランザクションIDと、前記トランザクションの処理経路における前記各オブジェクトの位置を示すカウンタ情報とを前記他のオブジェクトに伝え、
前記ログライタが記録した各イベントログには、対応するイベントに関わるトランザクションIDと、前記対応するイベントを実行したオブジェクトのプロセス名及び前記カウンタ情報と、前記対応するイベントの種別とが含まれている、オブジェクト間協調動作のログ管理方式。
【請求項2】　前記ログファイルが、個々のオブジェクトに割り当てられたオブジェクト別ログ領域を有する請求項1記載のログ管理方式。
【請求項3】　前記各オブジェクトのカウンタ情報には、前記トランザクションの処理経路における前記各オブジェクトの順位を示す番号が含まれる請求項1記載のログ管理方式。
【請求項4】　前記各オブジェクトのカウンタ情報には、前記トランザクションの処理経路における分岐の枝を識別する番号が含まれる請求項3記載のログ管理方式。
【請求項5】　少なくとも1つのノードに配置され、前記複数のノード内の前記ログファイルから、指定されたトランザクションIDをもったイベントログを収集するログコレクタを更に備えた請求項1記載のログ管理方式。
【請求項6】　複数のノードを有する通信ネットワークと、
前記複数のノードに分散して配置された複数のオブジェクトと、
各ノードに配置され、前記各ノード内の前記オブジェクトが実行したイベントのイベントログが格納されているログファイルと、
少なくとも1つのノードに配置され、指定されたトランザクションのイベントを追跡するイベントトラッカと、を備え、
前記複数のオブジェクトは、相互間で協調動作を依頼しながら1つのトランザクションを連携して処理することができ、
各オブジェクトは、他のオブジェクトに協調動作を依頼するとき、前記トランザクションを識別するためのトランザクションIDと、前記トランザクションの処理経路における前記各オブジェクトの位置を示すカウンタ情報とを前記他のオブジェクトに伝え、
前記ログファイル内の各イベントログには、対応するイベントに関わるトランザクションIDと、前記対応するイベントを実行したオブジェクトのプロセス名及び前記カウンタ情報と、前記対応するイベントの種別とが含まれており、
前記各イベントトラッカはログコレクタを有し、このログコレクタは前記複数のノード内の前記ログファイルから、指定されたトランザクションIDをもったイベントログを収集する、オブジェクト間協調動作のイベント追跡方式。
【請求項7】　前記イベントトラッカは更にログアナライザを有し、このログアナライザは、前記ログコレクタが収集した前記イベントログに含まれる前記プロセス名、前記カウンタ情報、及び前記イベント種別に基づいて、前記トランザクションの経過を解明してユーザに提示する、請求項6記載のイベント追跡方式。
【請求項8】　各ノードに前記イベントトラッカが配置されている請求項6又は7記載のオブジェクト間協調動作のイベント追跡方式。
【請求項9】　前記イベントトラッカが更にログライタを有し、このログライタは前記各ノード内の前記オブジェクトが実行したイベントのイベントログを前記ログファイルに記録する請求項8記載のイベント追跡方式。
【請求項10】　前記通信ネットワークは複数の物理ネットワークを含み、
各物理ネットワークの所定の少なくとも1つのノードに配置されたログコレクションエージェントを更に備え、
前記各物理ネットワークの各ログコレクションエージェントは、ログ収集依頼を受けて、同じ物理ネットワーク内の全てのノード内のログファイルから、前記指定されたトランザクションIDをもったイベントログを収集して、収集したイベントログを前記ログ収集依頼の依頼元へ送る、請求項6記載のイベント追跡方式。
【請求項11】　前記各物理ネットワークの各ログコレクションエージェントは、前記ログ収集依頼を受けて、他の物理ネットワーク内の別のログコレクションエージェントへ更なるログ収集依頼を送り、そして、前記別のログコレクションエージェントが収集したイベントログを受けて、受けたイベントログを前記ログ収集依頼の依頼元へ送る、請求項10記載のイベント追跡方式。
【請求項12】　前記各オブジェクトのカウンタ情報には、前記トランザクションの処理経路における前記各オブジェクトの順位を示す番号が含まれる請求項6記載の

ログ追跡方式。

【請求項13】　前記各オブジェクトのカウンタ情報には、前記トランザクションの処理経路における分岐の枝を識別する番号が含まれる請求項12記載のログ追跡方式。

【請求項14】　複数のノードを有する通信ネットワークと、前記複数のノードに分散して配置された複数のオブジェクトと、各ノードに配置されたログファイルとを備え、前記複数のオブジェクトは、相互間で協調動作を依頼しながら1つのトランザクションを連携して処理することができる環境において、

或るオブジェクトから他のオブジェクトに協調動作を依頼するとき、前記トランザクションを識別するためのトランザクションIDと、前記トランザクションの処理経路における前記或るオブジェクトの位置を示すカウンタ情報とを、前記或るオブジェクトから前記他のオブジェクトに伝えるステップと、

前記各ノード内の前記オブジェクトが実行したイベントのイベントログを、前記各ノード内の前記ログファイルに記録するステップと、を有し、

各イベントログには、対応するイベントに関わるトランザクションIDと、前記対応するイベントを実行したオブジェクトのプロセス名及び前記カウンタ情報と、前記対応するイベントの種別とが含まれている、オブジェクト間協調動作のログ管理方法。

【請求項15】　前記複数のノード内の前記ログファイルから1つのノードへ、指定されたトランザクションIDをもったイベントログを収集するステップを更に有した請求項16記載のログ管理方法。

【請求項16】　複数のノードを有する通信ネットワークと、前記複数のノードに分散して配置された複数のオブジェクトと、各ノードに配置された、前記各ノード内の前記オブジェクトが実行したイベントのイベントログが格納されているログファイルとを備え、前記複数のオブジェクトは、相互間で協調動作を依頼しながら1つのトランザクションを連携して処理することができ、各イベントログには、対応するイベントに関わるトランザクションIDと、前記対応するイベントを実行したオブジェクトのプロセス名及び前記カウンタ情報と、前記対応するイベントの種別とが含まれている環境において、

指定されたトランザクションのイベントを追跡するステップを有し、

この追跡ステップは、前記複数のノード内の前記ログファイルから、指定されたトランザクションIDをもったイベントログを収集するステップを含む、オブジェクト間協調動作のイベント追跡方法。

【請求項17】　前記追跡ステップは、前記収集ステップで収集した前記イベントログに含まれる前記プロセス名、前記カウンタ情報、及び前記イベント種別に基づいて、前記トランザクションの経過を解明してユーザに提示するステップを含む、請求項16記載のイベント追跡方法。

【請求項18】　前記各ノード内の前記オブジェクトがイベントを実行したとき、実行したイベントのイベントログを前記各ノード内の前記ログファイルに記録するステップを更に有する請求項17記載のイベント追跡方法。

【請求項19】　複数のノードを有する通信ネットワークと、前記複数のノードに分散して配置された複数のオブジェクトと、各ノードに配置されたログファイルとを備え、前記複数のオブジェクトは、相互間で協調動作を依頼しながら1つのトランザクションを連携して処理することができ、各オブジェクトは、他のオブジェクトに協調動作を依頼するとき、前記トランザクションを識別するためのトランザクションIDと、前記トランザクションの処理経路における前記各オブジェクトの位置を示すカウンタ情報とを前記或るオブジェクトから前記他のオブジェクトに伝える環境において、

前記各ノードに配置され、

前記各ノード内のオブジェクトが実行したイベントのイベントログであって、前記イベントに関わるトランザクションIDと、前記オブジェクトのプロセス名及び前記カウンタ情報と、前記イベントの種別とが含まれている前記イベントログを、前記各ノード内の前記ログファイルに記録するログライタ。

【請求項20】　複数のノードを有する通信ネットワークと、前記複数のノードに分散して配置された複数のオブジェクトと、各ノードに配置されたログファイルとを備え、前記複数のオブジェクトは、相互間で協調動作を依頼しながら1つのトランザクションを連携して処理することができ、各オブジェクトは、他のオブジェクトに協調動作を依頼するとき、前記トランザクションを識別するためのトランザクションIDと、前記トランザクションの処理経路における前記各オブジェクトの位置を示すカウンタ情報とを前記或るオブジェクトから前記他のオブジェクトに伝える環境において、

前記各ノード内のオブジェクトが実行したイベントのイベントログであって、前記イベントに関わるトランザクションIDと、前記オブジェクトのプロセス名及び前記カウンタ情報と、前記イベントの種別とが含まれている前記イベントログを、前記各ノード内の前記ログファイルに記録するログライタとして、前記各ノードを動作させるためのコンピュータプログラムを担持したコンピュータ読み取り可能な記録媒体。

【請求項21】　複数のノードを有する通信ネットワークと、前記複数のノードに分散して配置された複数のオブジェクトと、各ノードに配置された、前記各ノード内の前記オブジェクトが実行したイベントのイベントログが格納されているログファイルとを備え、前記複数のオブジェクトは、相互間で協調動作を依頼しながら1つの

トランザクションを連携して処理することができ、各イベントログには、対応するイベントに関わるトランザクションＩＤと、前記対応するイベントを実行したオブジェクトのプロセス名及び前記カウンタ情報と、前記対応するイベントの種別とが含まれている環境において、少なくとも一つのノードに配置され、前記複数のノード内の前記ログファイルから、指定されたトランザクションＩＤをもったイベントログを収集するログコレクタを有して、収集されたイベントログを頼りに指定されたトランザクションのイベントを追跡するイベントトラッカ。

【請求項２２】　前記ログコレクタが収集した前記イベントログに含まれる前記プロセス名、前記カウンタ情報、及び前記イベント種別に基づいて、前記トランザクションの経過を解明してユーザに提示するログアナライザを更に有する請求項２１記載のイベントトラッカ。

【請求項２３】　各ノードの配置され、前記各ノード内の前記オブジェクトがイベントを実行したとき、実行したイベントのイベントログを前記各ノード内の前記ログファイルに記録するログライタを更に有した請求項２１記載のイベントトラッカ。

【請求項２４】　複数のノードを有する通信ネットワークと、前記複数のノードに分散して配置された複数のオブジェクトと、各ノードに配置された、前記各ノード内の前記オブジェクトが実行したイベントのイベントログが格納されているログファイルとを備え、前記複数のオブジェクトは、相互間で協調動作を依頼しながら１つのトランザクションを連携して処理することができ、各イベントログには、対応するイベントに関わるトランザクションＩＤと、前記対応するイベントを実行したオブジェクトのプロセス名及び前記カウンタ情報と、前記対応するイベントの種別とが含まれている環境において、前記複数のノード内の前記ログファイルから、指定されたトランザクションＩＤをもったイベントログを収集するログコレクタを有して、収集されたイベントログを頼りに指定されたトランザクションのイベントを追跡するイベントトラッカとして、前記各ノードを機能させるためのコンピュータプログラムを担持したコンピュータ読み取り可能な記録媒体。

【請求項２５】　前記ログコレクタが収集した前記イベントログに含まれる前記プロセス名、前記カウンタ情報、及び前記イベント種別に基づいて、前記トランザクションの経過を解明してユーザに提示するログアナライザを、前記イベントトラッカが更に有する請求項２４記載の記録媒体。

【請求項２６】　前記各ノード内の前記オブジェクトがイベントを実行したとき、実行したイベントのイベントログを前記各ノード内の前記ログファイルに記録するログライタを、前記イベントトラッカが更に有する請求項２４記載の記録媒体。

【発明の詳細な説明】

【０００１】

【発明の技術分野】本発明は、通信ネットワーク上に散在する複数のオブジェクト間で非同期協調動作が行われる環境におけるログ管理およびイベント追跡のための方式および方法に関する。

【０００２】

【従来の技術】ネットワーク上で複数のオブジェクトが協調動作を行う環境において、従来は各マシン（ノード）単位で個別に通信ログが取得されている。

【０００３】

【発明が解決しようとする課題】しかし、単純に各マシン単位で通信ログを取得する従来技術においては、或るマシン上の通信ログを頼りに、協調動作を行った他の全てのマシンのログを探し出して回収することが困難である。そのため、特に不特定多数のオブジェクト間で非同期協調動作が行われる一層高度な分散環境においては、トランザクションのイベントを追跡して障害解析やデバッグを行うことが難しい。

【０００４】従って、本発明の目的は、オブジェクト間の協調動作に関するログ回収やイベント追跡を容易にすることにある。

【０００５】

【課題を解決するための手段】本発明の第１の側面に従うオブジェクト間協調動作のログ管理方式は、複数のノードを有する通信ネットワークと、複数のノードに分散して配置された複数のオブジェクトと、各ノードに配置されたログファイルと、各ノードに配置され、各ノード内のオブジェクトが実行したイベントのイベントログをログファイルに記録するログライタとを備える。複数のオブジェクトは、相互間で協調動作を依頼しながら１つのトランザクションを連携して処理することができる。各オブジェクトは、他のオブジェクトに協調動作を依頼するとき、トランザクションを識別するためのトランザクションＩＤと、そのトランザクションの処理経路における各オブジェクトの位置を示すカウンタ情報とを他のオブジェクトに伝える。また、ログライタが記録した各イベントログには、対応するイベントに関わるトランザクションＩＤと、対応するイベントを実行したオブジェクトのプロセス名及びカウンタ情報と、対応するイベントの種別とが含まれている。

【０００６】このログ管理方式によれば、１つのトランザクションのイベントログは複数のノードのログファイルに分散して記録される。しかし、それらのイベントログにはトランザクションＩＤが含まれているため、このトランザクションＩＤを頼りに、特定のトランザクションのイベントログを検索することが可能である。そして、検索したログに含まれているプロセス名、カウンタ情報及びイベント種別に基づいて、そのトランザクションのイベントの相互関係が把握できるので、トランザク

ションの経過を解明することが可能である。

【0007】本発明の第2の側面に従うオブジェクト間協調動作のイベント追跡方式は、複数のノードを有する通信ネットワークと、複数のノードに分散して配置された複数のオブジェクトと、各ノードに配置され、各ノード内のオブジェクトが実行したイベントのイベントログが格納されているログファイルと、少なくとも1つのノードに配置され、指定されたトランザクションのイベントを追跡するイベントトラッカとを備える。複数のオブジェクトは、相互間で協調動作を依頼しながら1つのトランザクションを連携して処理することができる。各オブジェクトは、他のオブジェクトに協調動作を依頼するとき、トランザクションIDと、そのトランザクションの処理経路における各オブジェクトの位置を示すカウンタ情報とを、他のオブジェクトに伝える。ログファイル内の各イベントログには、対応するイベントに関わるトランザクションIDと、対応するイベントを実行したオブジェクトのプロセス名及びカウンタ情報と、対応するイベントの種別とが含まれる。各イベントトラッカはログコレクタを有しており、このログコレクタは複数のノード内のログファイルから、指定されたトランザクションIDをもったイベントログを収集することができる。

【0008】このイベント追跡方式によれば、イベントトラッカは、トランザクションIDを頼りに特定のトランザクションのイベントログをネットワークから収集することができる。そして、収集した検索したログに含まれているプロセス名、カウンタ情報及びイベント種別に基づいて、イベントトラッカは、そのトランザクションのイベントを追跡することができる。

【0009】望ましくは、前記イベントトラッカは更にログアナライザを有することができる。このログアナライザは、収集されたイベントログに含まれるプロセス名、カウンタ情報、及びイベント種別に基づいて、トランザクションの経過を解明してユーザに提示する。

【0010】前記イベントトラッカは、ネットワーク上の全てのノードに配置してもよい。また、このイベントトラッカには、各ノード内のオブジェクトが実行したイベントのイベントログをログファイルに記録するログライタも含ませることができる。

【0011】望ましくは、各イベントトラッカのためにイベントログの収集を代行するログコレクションエージェントを、各物理ネットワーク毎に設けることができる。このログコレクションエージェントは、イベントトラッカからログ収集依頼を受けると、同じ物理ネットワーク内の全てのノード内のログファイルから、指定されたトランザクションIDをもったイベントログを収集し、更に、他の物理ネットワーク内の別のログコレクションエージェントへもログ収集依頼を送る。そして、このログコレクションエージェントは、同じ物理ネットワーク内の全てのノードから収集したイベントログと、別のログコレクションエージェントに収集してもらった他の物理ネットワーク上とを、依頼元のイベントトラッカに返送する。

【0012】上述したログライタ、イベントトラッカ、ログコレクタ、ログアナライザ、およびログコレクションエージェントは、典型的にはコンピュータがそれらのプログラムを実行することによって実現される。それらのプログラムは、各種のディスク型ストレージ、半導体メモリ、通信ネットワークなどの様々な媒体を通じてコンピュータにインストール又はロードすることができる。

【0013】

【発明の実施の形態】本発明の通信ログ管理及びイベント追跡方式の実施形態の説明に入る前に、この実施形態を適用することが有用な、新規の2種類のネットワーク環境について説明する。1つ目は新規なクライアント／サーバ環境であり、2つ目は、新規なマルチキャストのためのネットワーク環境である。

【0014】〔新規なクライアント／サーバ環境〕図1は、このクライアント／サーバ環境の全体構成を示す。この環境では、クライアントオブジェクトの知らない（つまり、クライアントオブジェクトにとって不特定の）1個以上のサーバオブジェクトが、そのクライアントオブジェクトと非同期で協調動作を行って或るトランザクションを遂行することができるようになっている。以下、詳細に説明する。

【0015】ネットワーク上に、少なくとも1つのクライアントオブジェクト（以下、単にクライアントという）1と、「サービスエージェント」と呼ばれる1セット又は複数セットのオブジェクトセット3、7と、複数のサーバオブジェクト（以下、単にサーバという）5A、5B、5C、…（これを纏めてサーバ群5ともいう）、別の1群以上のサーバ群9などが散在している。クライアント1は、原則として1つの特定のサービスエージェント3の傘下におかれている。サーバ群5も、原則として1つの特定のサービスエージェント3の傘下におかれ、同様に、別のサーバ群9も、原則として1つの別の特定のサービスエージェント7に傘下におかれている。ネットワークにおける各オブジェクトの場所は、ネットワーク上のノードを示すノード名と、ノード内のオブジェクト（プロセス）を示すプロセス名とにより特定される。

【0016】なお、図1では1つのクライアント1しか図示してないが、実際には、多くの他のクライアントも同じネットワーク上に存在しているのが普通である。しかし、1つのクライアント1に関する説明は他のクライアントにも同様にあてはまるので、他のクライアントについての図示および説明は省略する。

【0017】サービスエージェント3は、傘下の各サーバ5A、5B、5C、…のサービス名、場所及びステー

タスを知ってそれを管理している。別のサービスエージェント7も、傘下のサーバ群9に属する各サーバのサービス名、場所及びステータスを同様に管理している。一方、クライアント1は、ネットワーク上に散在するサーバ5A、5B、5C、…、9の個数、場所（ノード名、プロセス名）、ステータス（アイドル、処理中、故障などの現在状態）及び提供するサービス名ついて全く関知しない。

【0018】各オブジェクトの動作の概要は次の通りである。

【0019】各サーバ5A、5B、5C、…は、自己を傘下におさめるサービスエージェント3に対し、サービス開始時に自己のサービス名及び場所（ノード名とプロセス名）を知らせる（通信C1）。サービスエージェント3は、各サーバ5A、5B、5C、…から通知されたサービス名及び場所を記憶し、また、各サーバ5A、5B、5C、…のステータスも以後監視する。同様に、別のサーバ群9に属する各サーバもサービスエージェント7に自己のサービス名及び場所を通知し（通信C2）、それをサービスエージェント7が同様に管理する。

【0020】クライアント1は、サービス名を指定したサービス要求をサービスエージェント3に送る（通信C3）。サービスエージェント3は、サービス要求に応じられるサーバをサーバ群5内から探し、そのサーバ（例えばサーバ5Bとする）にサービス要求を送る（通信C4）。サーバ5Bは、要求されたサービスの処理を実行し、その処理結果をサービスエージェント3に返送する（通信C5）。サービスエージェント3は、その処理結果をクライアント1に送る（通信C6）。なお、この通信C6では、サービスエージェント3は、予めクライアント1から教えられているコールバック（CB）を用いてクライアント1を呼んで処理結果を渡す。

【0021】ところで、サービスエージェント3は、クライアント1からのサービス要求に応じられるサーバがサーバ群5内に無い場合、その旨をクライアント1に通知することもできるが、その代わりに、別のサービスエージェント7にサーバを照会することもできる（通信C7）。別のサービスエージェント7は、傘下のサーバ群9から要求に応じられるサーバを探し、そのサーバの場所をサービスエージェント3に通知する（通信C8）。サービスエージェント3は、その通知されたサーバにサービス要求を送り（通信C9）、その結果を受信し（通信C10）、それをクラアント1に送る（通信C6）。

【0022】以上のように、サービスエージェント3がクライアント1とサーバ5A、5B、5C、…間の通信を仲介することにより、クライアント1は、サーバとの通信に必要な情報（例えばサーバの個数や場所など）に関知することなしに、所望のサービス名を指定するだけでそのサービスの提供を受けることができる。つまり、クライアント1はサーバとの通信という問題から解放されるのである。このことにより、クライアント1のプログラム構築は、サーバの事情を考慮せずにすむため、非常に容易になる。

【0023】また、サービスエージェント3は、通信C6でクライアント1に処理結果を返すとき、所定のCBを用いてクライアント1を呼び出す。そのためクライアント1は、要求を送る通信C3から結果を受ける通信C6までの間ずっと待っている必要はなく、通信C3が終わり次第別の処理に移行することができる。つまり、オブジェクト間の非同期協調動作が可能である。

【0024】図2は、サービスエージェント3の構成と働きをより詳細に示している。

【0025】サービスエージェント3は、サービスエージェントのデーモンプロセスとして提供されるサービスマネージャ31と、サービスマネージャ31内に唯一存在するサービステーブル33と、クライアント1及びサーバ5A、5B、…の各々に組み込まれ又は付属したサービス・アプリケーションインタフェース（サービスAPI）35、37A、37B、37C、…とを含む。サービスマネージャ31は、サービスAPI35、37A、37B、…が知っている特定場所のノードにおかれる。各サービスAPI35、37A、37B、…は、当然、それぞれが組み込まれ又は付属したクライアント1及びサーバ5A、5B、5C、…と同じノードにおかれる。

【0026】サービスマネージャ31の主たる機能は、傘下の全てのサーバ5A、5B、…のサービス名、ノード名、プロセス名及びステータス（以下、これらを総称してサービス情報という）をサービステーブル33上で管理し、クライアント1からのサービス要求を受けると、サービステーブル33を参照して傘下のサーバ群5内から、そのサービス要求に応えられるサーバを傘下のサーバ群5から探すことである。

【0027】各サービスAPI35、37A、37B、…の主たる機能は、対応するクライアント1及びサーバ5A、5B、5C、…のために、サービスマネージャ31との通信や他のサーバ又はクライアントとの通信を代行することである。

【0028】図3及び図4は、図2に示した各オブジェクトの動作の流れを示す。また、図5は、サービステーブル31の内容例を示す。以下、図2～図5を参照して、各部の動作を説明する。

【0029】各サーバ5A、5B、…は、各々の起動時などに、各々のサービスAPI37A、37B、…に対しサービス提供宣言を送る（通信C21）。このサービス提供宣言には、それぞれのサーバのサービス処理を識別するためのサービス名と、所定の提供CBの指定とが含まれる。ここで、提供CBとは、各サーバ5A、5B、…内のサービス処理の実行主体であり、これを呼ぶことにより、各サーバ5A、5B、…にサービス処理を

実行させることができる。サービス提供宣言が各サービスAPI37A、37B、…に受け付けられると、その旨の確認応答が各サーバ5A、5B、…に返される。

【0030】各サービスAPI37A、37B、…は、各サーバ5A、5B、…からの上記サービス提供宣言に応答して、サービスマネージャ31に対し、各サーバのサービス名、ノード名およびプロセス名を含んだサービス提供宣言を送る（通信C22）。

【0031】サービスマネージャ31は、各サービスAPI37A、37B、…からサービス提供宣言を受けると、サービステーブル33に、宣言に含まれているサービス名、ノード名およびプロセス名を登録する（ステップS1）。この登録時、サービスマネージャ31は、サーバの状態を示すテーブル33上のステータスを「アイドル」に設定する。従って、サービステーブル33には、図5に例示するように、既に起動した全てのサーバ5A、5B、…のサービス名、ノード名、プロセス名およびステータスが記録されることになる。

【0032】クライアント1は、自己のサービスAPI35に対しサービス要求を送る（通信C23）。このサービス要求には、要求するサービスを識別するためのサービス名と、所定の応答CBの指定とが含まれる。ここで、応答CBとは、クライアント1内のサービス処理結果を受信する主体であり、これを呼ぶことによりクライアント1にサービス処理結果を受け取らせることができる。サービス要求がサービスAPI35に受け付けられると、その旨の確認応答がサービスAPI35からクライアント1に返される。この確認応答を受けた後は、サービス処理の結果が来ればサービスAPI35が応答CBを呼んでくれるので、クライアント1はサービス処理の結果を待つことなく、別の処理に移行することができる。

【0033】サービスAPI35は、クライアント1からサービス要求を受けると、サービス名をキーとして、サービス情報の照会をサービスマネージャ31に対して行う（通信C24）。

【0034】サービスマネージャ31は、サービステーブル33を参照して、そのサービス名を現在提供することができるサーバを探す（ステップS2）。具体的には、テーブル33上のサービス名が照会にかかるサービス名と一致し、かつ、テーブル33上のステータスが「アイドル」であるサーバを探す。この条件を満たすサーバが見つかれば、サービスマネージャ31は、そのサーバの場所を示すノード名とプロセス名をテーブル33から読み出し、照会元のサービスAPI35に返送する（通信25）。

【0035】サービスAPI35は、サービスマネージャ31から受け取ったノード名とプロセス名とを宛先にして（例えば、サーバ3Aとする）、クライアント1の要求したサービス名を含んだサービス要求を送信する（通信C26）。

【0036】サーバ3Aのサービス API37Aは、サービス要求を受信すると、予めサーバ3Aから指定されている提供CBを呼ぶ（通信C27）。これにより、サーバ3Aのサービス処理が開始される。サービス処理が開始されると、サービスAPI37Aは、ステータスを「処理中」に変更する旨のステータス変更通知をサービスマネージャ31に送る（通信C28）。このステータス変更通知には、サーバ3Aのノード名及びプロセス名が含まれている。サービスマネージャ31は、そのステータス変更通知を受けると、そのノード名及びプロセス名に該当するサービステーブル33上のステータスを「アイドル」から「処理中」に変更する（ステップS3）。

【0037】サービス処理が終わって提供CBが終了すると、サーバ3Aは提供CBの終了宣言をサービスAPI37Aに送る（通信C29）。この終了宣言には、実行したサービス処理の結果（例えば、データベース検索を行った場合の検索結果など）が含まれている。

【0038】サービスAPI37Aは、提供CB終了宣言を受けると、ステータスを「アイドル」に変更する旨のステータス変更通知をサービスマネージャ31に送る（通信C30）。サービスマネージャ31は、その通知を受けて、サービステーブル33上の該当するステータスを「処理中」から「アイドル」に変更する（ステップS4）。

【0039】提供CB終了宣言を受けたサービスAPI37Aは、さらに、サービス処理結果をサービス要求元のクライアント1に宛てて返送する（通信C31）。クライアント1のサービスAPI35は、そのサービス処理結果を受信すると、応答CBを呼ぶ（通信C32）。これにより、クライアント1は、サービス処理結果を受け取る。

【0040】ところで、サービスマネージャ31は、クライアント1のサービスAPI35からサービス情報照会を受けたとき（通信C24）、条件を満たすサーバがサービステーブル33から見つからなかった場合、図2に示すように、別のサービスエージェントのサービスマネージャに対して、同内容のサービス情報照会を送ることができる（通信C33）。すると、その別のサービスマネージャは、自己のサービステーブルから条件を満たすサーバを探して、そのノード名とプロセス名を照会元のサービスマネージャ31に返送する（通信C34）。サービスマネージャ31は、そのノード名とプロセス名とをクライアント1のサービスAPI35に送り（通信25）、サービスAPI35はそのノード名とプロセス名を宛先にしてサービス要求を発し（通信C35）、そしてその結果を受けることができる（通信C36）。

【0041】以上のような具体的な構成及び動作により、図1を参照して既に説明した通りの、クライアント

1はサーバを意識する必要が無い、オブジェクト間の非同期協調動作が可能である、という利点が得られる。

【0042】図6は、オブジェクト間非同期協調動作が可能であることによって得られる一つの具体的な利点の例を示している。

【0043】例えば、次のようなケースを想定する。すなわち、クライアント1は、特定の計算をなるべく速く行うことを欲している。その特定の計算を行う方法にはアルゴリズムAとBの2種類があり、いずれの方法がより速いかは、入力データの値に依存する。アルゴリズムA又はBを実装したサーバ5A、5B、5C、…がネットワーク上に散在しており、それぞれのアルゴリズムについて図示のように多重にサーバが存在している。

【0044】このようなケースでは、クライアント1はサービスエージェント3に対し、「アルゴリズムAによる計算要求」と「アルゴリズムBによる計算要求」という2つのサービス要求をたて続けに発することができる（通信C41、C42）。非同期処理が可能であるから、一方の計算が完了する前に、他方の計算要求を発することができるからである。

【0045】すると、サービスエージェント3は、即座に、アルゴリズムA、Bの各々についてステータスがアイドルのサーバ（例えば、サーバ5Aとサーバ5Cとする）を見つけ、それらに対し、その特定の計算を依頼する（通信C43、C44）。サーバ5Aとサーバ5CはアルゴリズムAとBでそれぞれ計算を行い、いずれか早く計算を完了した方（例えばサーバ5A）が先に、計算結果を返す（通信C45）。従って、クライアント1は、労せずして、速いほうの計算結果を受け取ることができる（通信C46）。

【0046】なお、上記説明では、個々のクライアントおよび個々のサーバは、1つのサービスエージェントの傘下におかれており、その1つのサービスエージェントとのみ通信している。しかし、必ずしもそうである必要はなく、或るクライアント又は或るサーバが、2つ以上のサービスエージェントの傘下におかれていてもよい。その場合、そのクライアントのサービスAPI又はそのサーバのサービスAPIは、自分が傘下にある2つ以上のサービスエージェントのサービスマネージャの場所を知っている必要がある。こうすることにより、サービスマネージャの負荷分散を図ることが可能である。また、例えば、或るサービスエージェントは或る特定種類のサービスを提供するサーバ群を傘下におさめ、別のサービスエージェントはまた別の特定種類のサービスを提供するサーバ群を傘下におさめているような場合、双方のサービスを受けたい1つのクライアントが、その2つのサービスエージェントの傘下に入って、ほしいサービスに応じてサービスエージェントを使い分けるといった構成も可能である。

【0047】ところで、上記説明では、クライアントやサーバのサービスAPIを、サービスエージェントの一部という位置づけで説明したが、これは一つの説明の方法又は概念化の方法に過ぎないものであって、別の説明も可能であることに注意されたい。すなわち、例えば、各クライアントや各サーバのサービスAPIを、サービスエージェントの一部ではなく、各クライアントや各サーバの一部という位置づけで理解することもできるし、或いは、サービスエージェント、クライアント及びサーバのいずれにも属さない独立したオブジェクトとして理解することも可能である。

【0048】〔新規なマルチキャスト環境〕図7は、このマルチキャスト環境の全体構成を示す。この環境では、メッセージの送信元のオブジェクトが指定する特定のドメインに参加してはいるが、送信元オブジェクトが具体的に知ってはいない（つまり、送信元オブジェクトにとって不特定の）複数のオブジェクトにたいして、メッセージをマルチキャストできるよになっている。以下、詳細に説明する。

【0049】図7に示すように、或る通信ネットワーク100が、ルータ130-1、130-2、…などによって相互接続された多数の物理的なネットワークセグメントの集合体として構成されている。この通信ネットワーク100上に、1以上の論理ネットワーク110-1、110-2、…が定義されている。各論理ネットワーク110-1、110-2、…には、通常、多数のノード（=ホスト）120-1、120-2、…が含まれている。

【0050】ノード120-1、120-2、…の各々は、1つ又はそれ以上の数のアプリケーション（AP）オブジェクト（=プロセス）140-1、140-2、…を有している（図7では、ノード120-1以外の他のノード内のAPオブジェクトの図示は省略してある）。APオブジェクト140-1、140-2、…は、それぞれ特定の業務に関わる処理を行うが、業務の種類又は内容に応じて予め定義された幾つかの業務グループのいずれかに参加している。例えば、APオブジェクト140-1は特定のデータベースの検索を行う業務グループに参加し、別のAPオブジェクト140-2は計算を行う業務グループに参加する、などというようにである。個々の業務グループはドメインと呼ばれる。なお、本実施形態では業務の種類又は内容に応じてドメインを定義しているが、これは説明上の一例に過ぎない。別の観点（例えばコストの観点、管理形態の観点、処理速度の観点など）からドメインを定義しても勿論構わない。

【0051】図8は、このシステムの論理的な構成を示している。図中の菱形と黒ドットとを線で結んだシンボルは、菱形マーク側の要素1個に黒ドットマーク側の要素がN個（1つ以上）所属しているという「1対N」の所属関係を表している。

【0052】論理ネットワーク110は、物理ネットワークに設置されている多数のノード120の集合体である。この論理ネットワーク110は、論理的に定義されたものであり、物理的なセグメント構成やLAN/WANなどの定義とは無関係である。1つのノード120内では1つ以上のAPオブジェクト140が動作する。従って、各APオブジェクト140は、1つのネットワーク110と1つのノード120とに所属する。また、1つのドメイン170には、1つ以上のAPオブジェクト140が参加している。但し、各APオブジェクト140は、必ずしもドメイン170に参加する必要はない。つまり、何のドメインにも属さないAPオブジェクト140が存在してもよい。

【0053】再び図7を参照する。各論理ネットワーク110-1、110-2、…では、所定の1つのノードに、ネットワークアクタ（NWA）と呼ばれるデーモンプロセスが配置される。例えば、論理ネットワーク110-1では、ノード120-2にネットワークアクタ150-1が配置され、論理ネットワーク110-2では、ノード120-6にネットワークアクタ150-2が配置されている。このネットワークアクタ150-1、150-2、…は、それぞれの論理ネットワーク110-1、110-2、…における全体のメッセージ配信制御を行うものである。

【0054】また、個々のノード120-1、120-2、…には、ノードアクタ（NDA）と呼ばれるデーモンプロセス160-1、160-2、…が1つづつ配置されている。このノードアクタ160-1、160-2、…は、それぞれのノード120-1、120-2、…内でのメッセージ配信制御を行うものである。

【0055】図9は、ネットワークアクタ150とノードアクタ160の機能を具体的に示している。

【0056】特定のドメインに参加しているAPオブジェクト群へメッセージをマルチキャストしたいAPオブジェクト140は、自分の所属する論理ネットワーク上のネットワークアクタ150に配信の都度テンポラリなパス181を張って、そのメッセージを送出する。送出後、パス181は直ちに切断される。

【0057】ネットワークアクタ150とそれが所属する論理ネットワーク110上の個々のノードアクタ160との間には、そのノードアクタ160の起動（典型的にはマシンブート）タイミングで、配信パス183が張られる。そして、そのノードアクタ160の終了（典型的にはマシンのシャットダウン）タイミングで、それぞれの配信パス183は切断される。そのノードアクタ160の動作中、その配信パス183は張られたままである。

【0058】ネットワークアクタ150は、それが所属する論理ネットワーク110上の稼動中のノード120を管理するためのノードアクタテーブル151を、そのネットワークアクタ150のメモリ上に有している。このノードアクタテーブル151には、図10に例示するように、その論理ネットワーク110上の全ての稼動中のノード120のノード名と、そのノード120への配信パス183の識別番号（配信パスID、例えばファイルディスクリプタ）とが登録されている。個々のノードアクタ151が起動した時に、そのノードアクタ160が自分のノード名と配信パスとを示した起動通知をネットワークアクタ150へ送り、この起動通知に基づいて、ネットワークアクタ150が、その起動したノードアクタ160のノード名と配信パスIDとをノードアクタテーブル151に登録することができる。

【0059】特定ドメインへマルチキャストされるべきメッセージをAPオブジェクト140から受けたネットワークアクタ150は、ノードアクタテーブル151を参照して、その論理ネットワーク110上で稼働中の全てのノードアクタ160に対して、そのメッセージを配信する。

【0060】ノードアクタ160と、同じノード内の個々のAPオブジェクト140との間には、そのAPオブジェクト140の起動タイミングで配信パス185が張られる。そして、そのAPオブジェクト140の終了タイミングで、その配信パス185は切断される。そのAPオブジェクト140の動作中、その配信パス185は張られたままである。

【0061】ノードアクタ160は、それと同じノード上の稼働中のAPオブジェクト140を管理するためのAPオブジェクト管理テーブル161及びドメインテーブル163を有している。APオブジェクトテーブル161には、図11に例示するように、そのノード内で稼働中の全てのAPオブジェクト140のオブジェクト名と、その稼働中のAPオブジェクト140への配信パス185の配信パスIDとが登録されている。ドメインテーブル163には、図12に例示するように、その稼働中の全てのAPオブジェクト140のオブジェクト名と、その稼働中のAPオブジェクト140が参加しているドメインの名称（ドメイン名）とが登録されている。個々のAPオブジェクト140が起動した時に、そのAPオブジェクト140が自分のオブジェクト名と配信パスとドメイン名とを示した起動通知をノードアクタ160へ送り、この起動通知に基づいて、ノードアクタ160が、その起動したAPオブジェクト140のオブジェクト名と配信パスIDとをAPアプリケーションテーブル161とドメインテーブル163とに登録することができる。

【0062】特定のドメインへマルチキャストされるべきメッセージをネットワークアクタ150から受けたノードアクタ160は、APオブジェクトテーブル161とドメインテーブル163とを参照して、指定されたドメインに参加している稼働中のAPオブジェクト140

に対してのみ、そのメッセージを配信する。

【0063】図13は、APオブジェクト140から送出されるメッセージのフォーマットを示している。

【0064】メッセージ190のパケットには、通信制御のための情報を表すヘッダ191と、メッセージの正味の内容であるユーザデータ193とが含まれている。ヘッダ190には、送信元のAPオブジェクト140を示すオブジェクト名、ネットワーク名、ノード名及びドメイン名と、送信先のAPオブジェクト名140を示すオブジェクト名、ネットワーク名、ノード名及びドメイン名とが含まれている。

【0065】特定のドメインに参加しているAPオブジェクト群にメッセージをマルチキャストしたいAPオブジェクト140は、その特定のドメインの名称を、メッセージ190のヘッダ191の送信先ドメイン名にセットする。このとき、ヘッダ191の送信先オブジェクト名や送信先ノード名には、何のデータもセットしない（＝NULLデータをセットする）。このメッセージは、図14に例示するようにしてマルチキャストされる。

【0066】図14の例は、APオブジェクト140-1が、特定のドメインD1に参加しているAPオブジェクト群にメッセージをマルチキャストする場合を想定している。図中の矢印はメッセージの流れを示している。送信元のAPオブジェクト140-1は、自分の所属する論理ネットワーク上のネットワークアクタ150-1にテンポラリパス181を張って、そのネットワークアクタ150-1にメッセージを送信する。このメッセージのヘッダ191には、送信先ドメイン名として「D1」がセットされている。そのメッセージを受信したネットワークアクタ150-1は、同じ論理ネットワーク上で稼働中の全てのノードアクタ160-1、160-2、160-3に、そのメッセージを配信する。そのメッセージを受信した各ノードアクタ160-1、160-2、160-3は、同じノード内の稼働中のAPオブジェクトの中で、指定されたドメインD1に参加しているAPアプリケーション140-2、140-3、140-6に対してのみ、そのメッセージをそれぞれ配信する。

【0067】以上のようにして、特定のドメインに参加しているAPオブジェクトに対してのみメッセージがマルチキャストされる。その際、送信元のAPアプリケーション140-1は、1回だけメッセージを送信すればよい。論理ネットワーク上では、稼働中のノードに対してのみ、そのメッセージが送られる。ノード内では、稼働中で且つ指定されたドメインに参加しているAPオブジェクトに対してのみ、そのメッセージが送られる。このマルチキャスト方式によれば、従来のIPブロードキャストによる方法に比較して、トラヒックは少なくて済み、且つ、そのメッセージが不要なAPオブジェクトに余計な負担をかけることもない。さらに、論理ネットワークが物理的なセグメント構成とは無関係に、ルータやゲートウェイを跨いで定義できるから、ルータやゲートウェイを跨いだメッセージ配信が可能である。また、従来のユニキャストによる方法に比較して、ネットワークの構成やノードは一などの変更やAPオブジェクトの追加、削除に対してより柔軟に対応でき、且つ、休止中のノードやオブジェクトに無駄な送信を行うこともない。

【0068】上記では、送信元APオブジェクトが自分の所属する論理ネットワーク上の特定ドメインのオブジェクト群に対してマルチキャストする場合を説明したが、他のパターンのキャストも可能である。

【0069】例えば、メッセージヘッダ191において送信先ネットワーク名は指定するが、送信先のオブジェクト名、ノード名及びドメイン名は指定しないことにより、論理ネットワーク内の全ての稼働中APオブジェクトにメッセージを配信することができる。また、送信先のネットワーク名とノード名は指定するが、送信先のオブジェクト名及びドメイン名は指定しないことにより、指定したノード内の全ての稼働中APオブジェクトにメッセージを配信することができる。また、送信先のネットワーク名とノード名とドメイン名は指定するが、送信先のオブジェクト名は指定しないことにより、指定したノード内の指定したドメインに参加している稼働中APオブジェクトにのみメッセージを配信することができる。また、送信先のオブジェクト名とネットワーク名とノード名とを指定することにより、指定したノード内の指定した1つのオブジェクトにのみメッセージをおくすることができる。さらに、別の論理ネットワーク上のネットワークアクタの場所を知っているならば、その別の論理ネットワーク上のAPオブジェクトに対しても、上記のような種々のパターンでメッセージをキャストすることができる。

【0070】ところで、論理ネットワークはネットワークの物理的構成とは無関係に論理的に定義できるから、或る論理ネットワークが他の論理ネットワークと一部の領域で重なり合うように設定することもできる。すると、その重なり領域に存在するAPオブジェクトは複数の論理ネットワークに所属することになる。また、ドメインも自由に定義できるから、1つのAPオブジェクト140が複数のドメイン170に参加する（例えば、計算処理ドメインと高速処理ドメインの双方に参加する）こともあり得る。このようなケースにおいても本発明は問題無く適用することができる。

【0071】また、例えば、1つの広域の論理ネットワーク内に狭域の論理ネットワークを定義することもできる。その場合、それらのネットワークにそれぞれ1つづつネットワークアクタを配置し、広域のネットワークアクタと狭域のネットワークアクタとの間に配信パスを張り、広域のネットワークアクタが狭域のネットワークア

クタの配信パスIDを管理するというような、階層的なネットワークアクタのシステムを採用することができる。

【0072】〔本発明に従うログ管理及びイベント追跡方式の実施形態〕以下に、上述のような環境に好適な本発明に従う方式の一実施形態を説明する。なお、以下の説明には、上記環境におけるこの方式の具体的動作についての解説が含まれるが、それは例示に過ぎず、他の環境であってもこの方式が十分に動作し相応の効果を発揮し得る得ることを認識されたい。

【0073】図15にこの実施形態の全体を示す。

【0074】図15に示すように、通信ネットワーク200は、ルータ230－1、230－2、…などを介して相互接続された多数の物理的なネットワークセグメント（以下、物理ネットワークという）210－1、210－2、210－3、…に分けられる。それら物理ネットワーク210－1、210－2、210－3、…の各々では、所定の一つのノード（＝マシン）220－2、220－6、220－4、…に、ログコレクションエージェント（LCA）と呼ばれるデーモンプロセスのオブジェクト250－1、250－2、250－3、…が配置されている。また、各ノード220－1、220－2、…には、イベントトラッカ（ETR）と呼ばれるオブジェクト260－1、260－2、…が配置されている。

【0075】上述した環境では、各ノード220－1、220－2、…には、例えばアプリケーション（AP）オブジェクト（例えば、クライアント、サーバなど）またはデーモンプロセスのオブジェクト（例えば、サービスマネージャ、ネットワークアクタ、ノードアクタなど）のような、種々のオブジェクト240－1、240－2、…が含まれている。各イベントトラッカ（ETR）260－1、260－2、…は、各々のノードに含まれるそれらのオブジェクト240－1、240－2、…の通信ログを取得して各オブジェクト別にその通信ログを記録すること、および、或るトランザクションで通信障害のような問題が発生した場合、そのトランザクションに関係した全オブジェクトの通信ログを収集し分析して、そのトランザクションを構成するイベントを追跡することを行う。各ログコレクションエージェント（LCA）250－1、250－2、250－3、…は、各々の物理ネットワーク210－1、210－2、210－3、…内の各イベントトラッカ（ETR）260－1、260－2、…のために、ネットワーク200全体からの必要な通信ログの収集を代行する。

【0076】図16は、この実施形態の説明で用いる重要な幾つかの概念について説明している。図中、線で結んだ黒丸と菱形のシンボルは、菱形側の1つの概念が黒丸側のn個の概念から構成されるという「1対n」の所属関係を示しており、また、三角形のシンボルは、その三角形の頂点側の1つの概念の具体例として、その三角形の底辺側の複数の概念があるという「抽象－具体」の関係を示している。

【0077】図16に示すように、1つのトランザクション270は、複数のイベント280から構成される。トランザクション270とは、オブジェクト間の協調動作（この実施形態では非同期協調動作）の1単位であり、その具体例には、クライアントからサーバへサービスを要求しサーバがクライアントとにサービスを提供するという一連の処理である「サービス」271、および一つのオブジェクトから複数のオブジェクトへ同じメッセージがキャストされる一連の処理である「マルチキャスト」273などがある。イベント280とは、トランザクションの構成要素であり、その具体例には、「開始」281、「依頼」283、および「受付」385などがある（その詳細は後に説明する）。

【0078】図17は、イベントトラッカ260とログコレクションエージェント250に関してより詳細に示している。

【0079】1つのイベントトラッカ260－1は、ログコレクタ261、ログアナライザ263およびログライタ265という3種類のオブジェクトを有している（他の各イベントトラッカ260－2、260－3、…も同様である）。ログライタ265は、そのノード内の全てのオブジェクトのトランザクションに関わる動作を監視して各イベントのログを取得し、そのイベントログをそのノードのストレージ290内の2種類のイベントログファイル291又は293のいずれかに記録する。ログライタ265は、具体的には、各APオブジェクトにリンクされたライブラリ内のモジュール（例えば、図2に示すクライアント1やサーバ5のためのログを書くモジュールは、そのクライアント1やサーバ5にリンクされたサービスAPI37内に設けられる）、及び各デーモンプロセス内部のモジュール（例えば、図2に示したサービスマネージャ31のログを書くモジュールは、そのサービスマネージャ内部に設けられる）に分散して実装され得る。2種類のイベントログファイル291又は293の内の一方のイベントログファイル291は、APオブジェクトのイベントログを記録するためのものであり、他方のイベントログファイル293はデーモンプロセスのイベントログを記録するためのものである。なお、図17では、全てのノードに2種類のイベントログファイル291、293が1つづつ存在するかのように示されているが、必ずしもそうではない。例えば、APオブジェクトが存在しないノードにはAP用イベントログファイル291は存在せず、また、複数のデーモンプロセス（例えばノードアクタとネットワークアクタの双方）が存在するノードにはそれらに対応する複数のデーモンプロセス用イベントログファイル293が存在する。

【0080】図18および図19はそれぞれ、AP用イ

ベントログファイル291とデーモンプロセス用イベントログファイル293の構成を示している。

【0081】図18に示すように、AP用イベントログファイル291は、ノード名などが書かれたヘッダと、そのノードで起動できるAPオブジェクト（例えば、AP1、AP2およびAP3など）の最大個数分のAP別ログ領域295-1、295-2、…とを、各AP別ログ領域295-1、295-2、…は、メモリマッピングの方法によってそのノード内の個々のAPオブジェクトに割り当てられている。各AP別ログ領域295-1、295-2、…には、割り当てられたAPオブジェクトのプロセス名などが書かれたヘッダと、APオブジェクトのイベントログレコード297-1、297-2、…を書き込むための一定サイズの領域とがある。この一定サイズの領域にサイクリックに、新しいイベントログレコードが書き込まれていく。

【0082】図19に示すように、デーモンプロセス用イベントログファイル293には、割り当てられた特定のデーモンプロセス(例えば、ノードアクタ、ネットワークアクタ又はサービスエージェント)のプロセス名などが書かれたヘッダと、そのデーモンプロセスのイベントログレコード297-4、297-3、…を書き込むための領域とがある。この一定サイズの領域にサイクリックに、新しいイベントログレコードが書き込まれていく。

【0083】図20は、個々のイベントログレコード297の構成を示す。

【0084】イベントログレコード297は固定の長さを有し、そこには書き込み時刻、トランザクションID、中継点カウンタ、イベント種別などが記述される。トランザクションIDとは、個々のトランザクションに与えられるシステム内でユニークな識別番号である。中継点カウンタとは、トランザクションの処理経路において当該イベントを実行したオブジェクトが何番目のオブジェクトであるか（つまり、そのオブジェクトの順位）を示す数値である。マルチキャストのように複数のオブジェクト（枝）に処理経路が分岐するトランザクションでは、その枝を識別するための枝番が中継点カウンタに追加される（この場合、中継点カウンタ内のオブジェクト順位を数値を「幹番」とよぶ）。換言すれば、中継点カウンタは、そのイベント又はそのイベントを実行したオブジェクトの、トランザクション処理経路における位置を示す情報である。

【0085】イベント種別とは、図16に示した「開始」、「依頼」、「受付」などの具体的なイベント種類を識別するコードである。イベント種別には例えば次のようなものがある。

【0086】「STR」（start）：オブジェクト間の協調の「開始」である。中継点カウンタは「01」となる。

【0087】「PRO」（propose）：他のオブジェクトに対する協調の「依頼」である。マルチキャストでは、このイベントにおいて、中継点カウンタに枝番が追加される。

【0088】「ACC」（accept）：他のオブジェクトから依頼された協調を「受け付け」たことである。中継点カウンタは「PRO」に等しい。

【0089】「XCG」（exchange）：「変換」、つまり依頼された処理を行うという意味である。中継点カウンタはインクリメントされる。

【0090】「DIS」（discard）：協調の「放棄」、つまり、そのオブジェクトはこの先更なる協調を行なう必要が無い、または、依頼に対しての処理は行えないという意味である。

【0091】「END」（end）：協調の「終了」である。

【0092】図21および図22は、或るトランザクションにおけるイベントの発生とイベントログの書き込みの様子の具体例をそれぞれ示す。なお、これらの図及び以下の説明では、オブジェクトは「ノード名．オブジェクト名」の形式で、イベントは「中継点カウンタ．イベント種別」の形式で示す。また、枝番が追加された中継点カウンタは「幹番．第1の枝番．第2の枝番．第3の枝番」の形式で示す。図中の破線で囲いは、1つのトランザクションを意味する。

【0093】図21は、「サービス」のトランザクションの例を示しており、ここでは、クライアント「NC．AP1」がサービスマネージャ「NM．SMG」の仲介でサーバ「NS．AP2」から或るサービスを提供される。この種のトランザクションの動作の詳細は、既に図3および図4を参照して説明してある。なお、ログ処理では、クライアントやサーバに付属するAPIインタフェースは、それぞれのクライアント及びサーバの一部であるとされる。

【0094】最初に、クライアント「NC．AP1」において、開始イベント「01．STR」が発生し、続いて、サーバ照会のための依頼イベント「01．PRO」が発生する。この2つのイベントのログレコードが、ノード「NC」内のAP用イベントログファイル291内のクライアント「NC．AP1」に割り当てられた領域に書き込まれる。

【0095】サービスマネージャ「NM．SMG」では、サーバ照会の依頼に対応した受付イベント「01．ACC」が発生し、続いて、適当なサーバを探す変換イベント「02．XCG」が発生し、ここで中継点カウンタがインクレメントされる。続いて、サービスマネージャ「NM．SMG」では、クライアント「NC．AP1」（正確には、このクライアントに付属するAPIインタフェース）に照会結果を返答するための依頼イベント「02．PRO」が発生する。これら3つのイベント

のログレコードが、ノード「NM」内のサービスマネージャ用のイベントログファイル293に書き込まれる。

【0096】クライアント「NC．AP1」では、照会結果を受ける受付イベント「02．ACC」、サービス要求を生成する変換イベント「03．XCG」（ここで、中継点カウンタがインクレメントされる）及びサーバ「NS．AP2」へサービス要求を送る依頼イベント「03．PRO」が発生する。これらのイベントのログレコードが、ノード「NC」内のAP用イベントログファイル291に書き込まれる。

【0097】以下、サーバ「NS．AP2」、サービスマネージャ「NM．SMG」及びクライアント「NC．AP1」において、図3、4を参照して既に説明したような種々のイベントが発生し、それらイベントのログレコードが各ノード「NS」、「NM」、「NC」内のイベントログファイルに書き込まれる。

【0098】図22は、「マルチキャスト」のトランザクションの例を示しており、ここでは、メッセージが1つのAPオブジェクト「NS．AP1」から1つのネットワークアクタ「NN．NWA」へ送られ、そこから3つのノードへ送られ、それらのノード内でノードアクタ「N1．NDA」、「N2．NDA」、「N3．NDA」からAPオブジェクト「N1．AP2」、「N2．AP3」、「N3．AP6」へ送られる。この種のトランザクションの処理の詳細は、既に図9から図14を参照して詳細に説明してある。

【0099】まず、送信オブジェクト「NS．AP1」において、開始イベント「01．00．00．00．STR」が発生し、次いでネットワークアクタ「NN．NWA」へメッセージを発信する依頼イベント「01．01．00．00．PRO」が発生し、ここで第1の枝番「01」が幹番「01」の後に追加される。この2つのイベントのログレコードが、ノード「NS」内のAP用イベントログファイル291に書き込まれる。なお、依頼イベント「PRO」については、それが多重に発生したときに書き込むべきログレコードの個数を最小にするために、最初の依頼イベントと最後の依頼イベントのログレコードのみを書き込むこととする。従って、この送信オブジェクト「NS．AP1」では、1つの依頼イベント「01．01．00．00．PRO」のみが発生しているが、その1つの依頼イベント「01．01．00．00．PRO」のログレコードは最初の依頼イベント及び最後の依頼イベントの双方のログレコードとして、二重に書き込まれる。

【0100】ネットワークアクタ「NN．NWA」では、依頼イベント「01．01．00．00．PRO」に対応する受付イベント「01．01．00．00．ACC」が発生し、次いで、キャスト先のノードアクタを選択するといった変換イベント「02．01．00．00．XCG」が発生し、ここで幹番がインクレメントされる。次いで、ネットワークアクタ「NN．NWA」では、選択された3つのノードアクタ「N1．NDA」、「N2．NDA」、「N3．NDA」へそのメッセージをマルチキャストする3つの依頼イベント「02．01．01．00．PRO」、「02．01．02．00．PRO」（図示省略）及び「02．01．03．00．PRO」が発生し、その各々では、第2の枝番「01」、「02」、「03」が第1の枝番「01」の後に追加される。これらのイベントのログレコードが、このノード「NN」内のネットワークアクタ「NN．NWA」用のイベントログファイル293も書き込まれる。このとき、前述したように、発生した3つの依頼イベントのうち、最初の依頼イベント「02．01．01．00．PRO」と最後の依頼イベント「02．01．03．00．PRO」のログレコードだけが書き込まれる。

【0101】以下、選択された3つのノードアクタ「N1．NDA」、「N2．NDA」、「N3．NDA」、および選択された受信オブジェクト「N1．AP2」、「N2．AP3」、「N3．AP6」の各々において、所要のイベントが発生し、各イベントのログレコードがそれぞれのオブジェクト用のログ領域に書き込まれる。この過程で、中継点の番目が増えるに伴って、中継点カウンタでは、幹番がインクレメントされるだけでなく、より下位の枝番が追加されていく。

【0102】ところで、図21及び図22では図示を省略してあるが、トランザクションIDも、中継点カウンタのようなイベント自体の情報と一緒に、オブジェクト間で受け渡しされる。したがって、同じトランザクションに含まれる全てのイベントのログデータには、その同じトランザクションのIDが受け継がれる。

【0103】図21および図22に示した2つの例から分かるように、1つのトランザクションを構成する複数のイベントのログレコードは、そのトランザクションに関与した複数のノード内のイベントログファイルに分散して管理される。しかし、それら分散した複数のイベントログレコードは、それらに含まれる同じトランザクションIDによって論理的にリンクされ、そして、それらに含まれる中継点カウンタとイベント種別によってイベント間の相互関係が理解され得るようになっている。

【0104】再び図17を参照する。各ノードでは、普段は各イベントトラッカ260内のログライタ265が、上述したようにしてイベントログを取得しこれをイベントログファイル291、293に書き込んでいる。一方、或るトランザクションにおいて何らかの障害が発生したときには、その障害解析やデバッグを行うために、ユーザはその目的のトランザクションに関係した或るノード（典型的には、その目的トランザクションが発生したノード）のイベントトラッカのログコレクタ261を起動する。

【0105】ここで、例えば、図17に示すイベントトラッカ260－1のログコレクタ261が起動されたと仮定する。このログコレクタ261は、まず、そのノードのディスプレイ装置400に、検索キー入力画面を表示する（ステップS12）。この検索キー入力画面は、例えば図23に示すようなものである。ユーザはキーボードなどの入力装置300から、この検索キー入力画面に、検索キーとして、目的トランザクションを発生させたノード名、プロセス名および目的トランザクションの発生時刻（又は障害発生時刻）を入力することができる（ステップS13）。

【0106】検索キーが入力されそしてユーザより検索が命じられると、ログコレクタ261は、まず、その検索キーを用いて、自ノード内のイベントログファイル291、293内のイベントログをスキャンすることにより（ステップS14）、目的トランザクションの候補を抽出し、そして、そのトランザクション候補のリストをディスプレイ装置400に出力する（ステップS15）。そのトランザクション候補リストは例えば図24に示すようなもので、そこには、各トランザクション候補のトランザクションID、発生時刻、トランザクション種別などが示される。ユーザは、このリストの中から、入力装置300のマウス又はキーボードを用いて、目的トランザクションを決定することができる（S16）。

【0107】目的トランザクションが決定されると、ログコレクタ261は、自ノード内のイベントログファイル291、293から目的トランザクションのログファイルを検索する（S17）。更に、ログコレクタ261は、自分の物理ネットワーク210－1上のログコレクションエージェント250－1に対して、そのトランザクションに関わる他の全てのログファイルを収集することを依頼するメッセージを送る（S18）。すると、そのログコレクションエージェント250－1は、自分の物理ネットワーク210－1上の全てのノードのイベントトラッカ260－2、260－3、…に対して、その依頼メッセージをIPブロードキャストの方法で配信する（S19）。更に、ログコレクションエージェント250－1は、他の全ての物理ネットワーク210－2、…上の他のログコレクションエージェント250－2、…の各々に対して、その依頼メッセージをユニキャストの方法で個別に送る（ステップS20）。他のログコレクションエージェント250－2、…も、同様にして、自分の物理ネットワーク210－2、…上の全てのノードのイベントトラッカ220－6、220－7、…へ、その依頼メッセージをブロードキャストする（ステップS21）。

【0108】その依頼メッセージを受けた他のノードのイベントトラッカ260－2、260－3、…、260－6、260－7、…の各々では、各ログコレクタ261が各ノード内のイベントログファイル291、293から目的のトランザクションIDをもったイベントログレコードを検索し（ステップS22）、検索したレコードを依頼元である各ログコレクションエージェント250－1、250－2、…へ送信する（ステップS23）。これにより、各ログコレクションエージェント250－1、250－2、…には、各物理ネットワーク210－1、210－2上で多数のノードに分散していた目的トランザクションに関わるイベントログコードが収集される。各ログコレクションエージェント250－1、250－2、…は、収集されたイベントログコードを各依頼元へ送る（ステップS24、S25）。結果として、イベントトラッカ260－1のログコレクタ261は、目的トランザクションに関わる全てのイベントログレコードを収集する。

【0109】次に、イベントトラッカ260－1のログコレクタ261は、こうして収集したイベントログレコードを、リストの形でディスプレイする（ステップS26）と共に、ログアナライザ263に渡す（ステップS27）。ディスプレイされたイベントログレコードのリストは、例えば図25に示すようなものである。図25は、図22に示したトランザクションのログレコードを収集した場合を例示している。このレコードリストには、収集された各イベントログレコードの、トランザクションID、オブジェクト、中継点カウンタ、イベント種別およびが書き込み時刻などが示されている。なお、図25では、書き込み時刻を簡略的に「分」までしか示してないが、実際には各イベントの発生時刻を区別できる「ミリ秒」のオーダまで書き込まれている。なお、ログコレクタ261は、例えば、上述したログコレクタ261の殆どの機能を受け持つデーモンプロセスと、ログアナライザ263とのインタフェースを受け持つユーティリティ（コマンド）とのセットととして実装され得る。

【0110】ログアナライザ263は、それらのイベントログレコードの中継点カウンタとイベント種別とに基づいて、それらイベントを相互に関連付け、それによって、目的トランザクションにおいてどのオブジェクトがどのように連携動作したかという経過を明らかにし、その経過を表現したテキスト又はグラフィックスを編集してディスプレイ装置400に出力する。この経過表示画面は例えば図26に示すようなものであり、それにより、ユーザは目的トランザクションの経過を容易に把握することができる。なお、ログアナライザ263は、例えば、ユーティリティ（コマンド）として実装され得る。

【0111】以上、本発明の一実施形態を説明したが、本発明はこの実施形態にのみ限定されるものではなく、他の種々の形態でも実施することが可能である。例えば、本発明の方式は、オブジェクト間の非同期協調動作

だけでなく同期協調動作についても適用できる。

【図面の簡単な説明】

【図1】本発明の一実施形態の構成を示すブロック図。

【図2】同実施形態のより具体的な構成を示すブロック図。

【図3】同実施形態の動作を示すシーケンス図。

【図4】同実施形態の動作を示すシーケンス図。

【図5】サービステーブルの例を示す図。

【図6】オブジェクト間非同期協調動作による一つの利点を説明する図。

【図7】本発明の一実施形態の概略的な全体構成を示したブロック図。

【図8】同実施形態の概念的な構成を示したブロック図。

【図9】ネットワークアクタ150とノードアクタ160の機能を具体的に示したブロック図。

【図10】ノードアクタテーブルの一例を示した図。

【図11】APオブジェクトテーブルの一例を示した図。

【図12】ドメインテーブルの一例を示した図。

【図13】APオブジェクト140から送出されるメッセージの構成を示したデータフォーマット図。

【図14】特定ドメインへのメッセージ配信の階層を示したブロック図。

【図15】本発明に従うログ管理及びイベント追跡方式の一実施形態の全体構成を示すブロック図。

【図16】この実施形態の説明で用いる重要な幾つかの概念の説明図。

【図17】イベントトラッカ260とログコレクションエージェント250の構成および関係を示すブロック図。

【図18】AP用イベントログファイル291の構成を示す図。

【図19】デーモンプロセス用イベントログファイル293の構成を示す図。

【図20】イベントログレコード297の構成を示す図。

【図21】「サービス」におけるイベントの例を示すシーケンス図。

【図22】「マルチキャスト」におけるイベントの例を示すシーケンス図。

【図23】検索キー入力画面の例を示す図。

【図24】トランザクション候補リストの例を示す図。

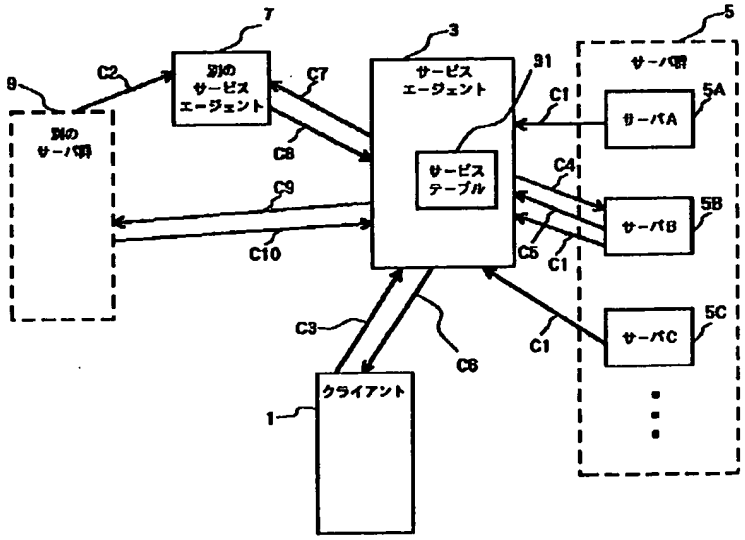
【図25】イベントログレコードリストの例を示す図。

【図26】経過表示画面の例を示す図。

【符号の説明】

200　通信ネットワーク
210　物理ネットワーク
220　ノード
250　ログコレクションエージェント
260　イベントトラッカ
270　トランザクション
280　イベント
261　ログコレクタ
263　ログアナライザ
265　ログライタ
291　アプリケーション用イベントログファイル
293　デーモンプロセス用イベントログファイル
297　イベントログレコード

【図1】

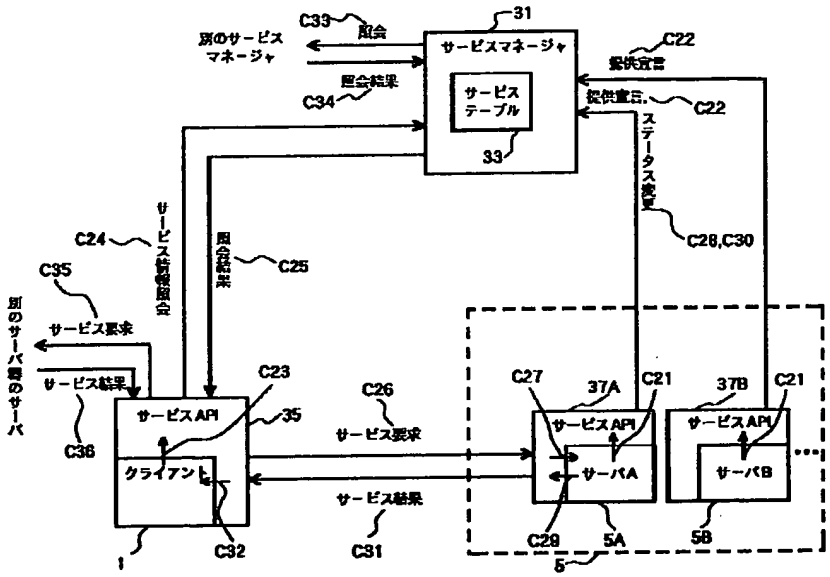


【図10】

ノード名	配信パスID
ND 1	5
ND 2	7
ND 3	8
⋮	⋮

151

【図2】

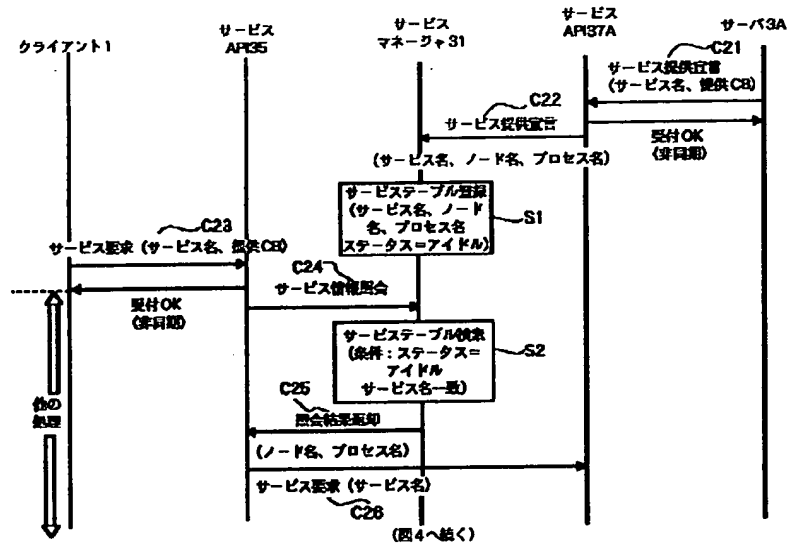


【図11】

オブジェクト名	配信パスID
AP 1	4
AP 2	9
AP 3	5
⋮	⋮

161

【図3】

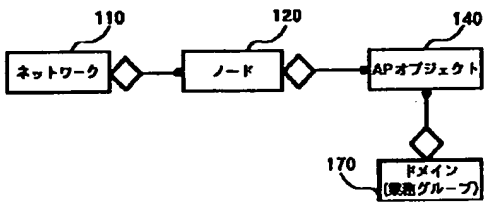


【図12】

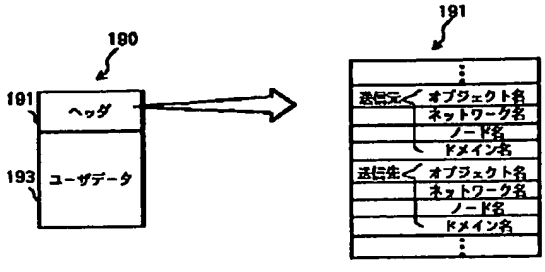
オブジェクト名	ドメイン名
AP 1	D1
AP 2	D1
AP 3	-
⋮	⋮

163

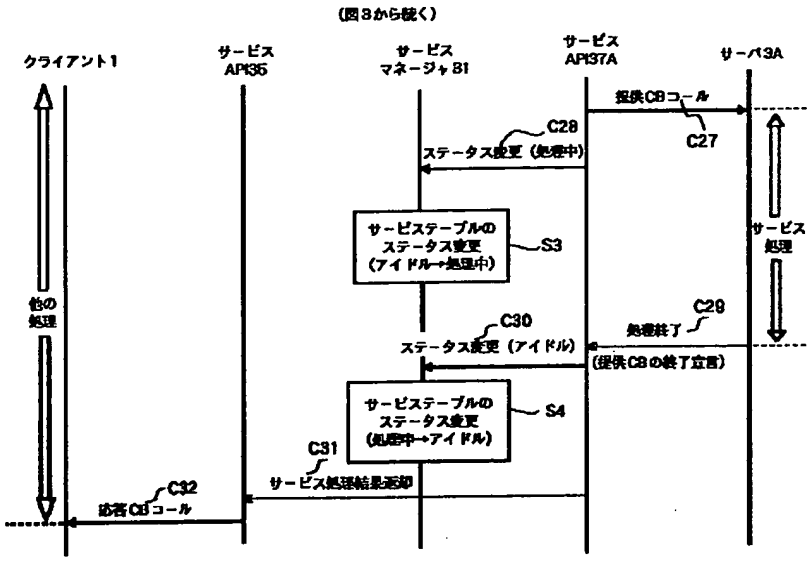
【図8】



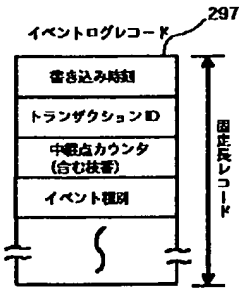
【図13】



【図4】



【図20】

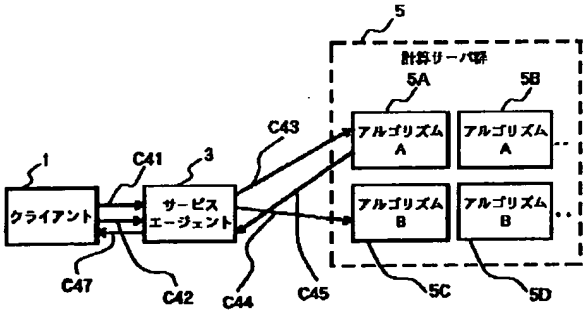


【図5】

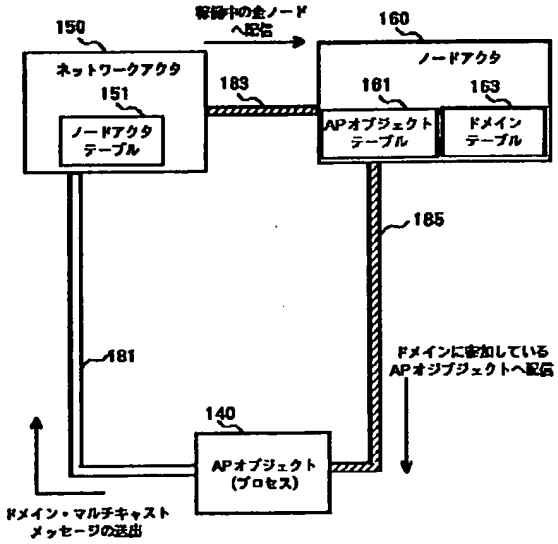
33
サービステーブル

サービス名	ノード名	プロセス名	ステータス
XX検索	node A	prc 1	処理中
YY検索	node B	prc 1	アイドル

【図6】



【図9】

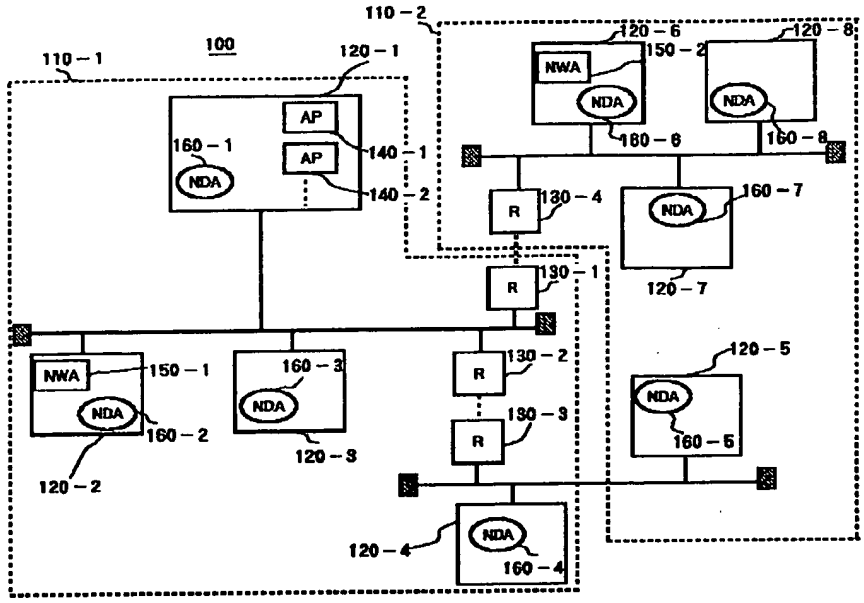


【図24】

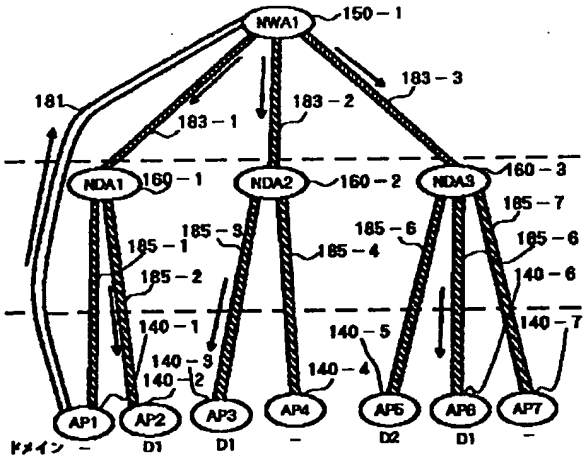
トランザクション設備

ID		発生時刻	種別
578	:	07/29 16:05	サービス
579	:	07/29 16:45	マルチキャスト
⋮		⋮	⋮

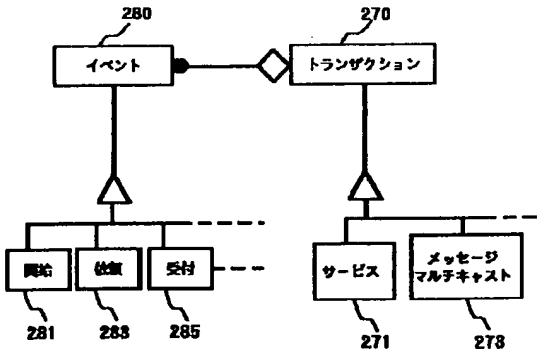
【図7】



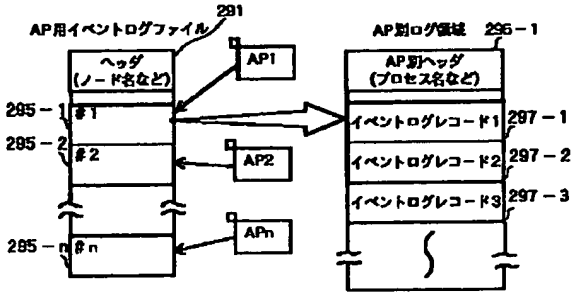
【図14】



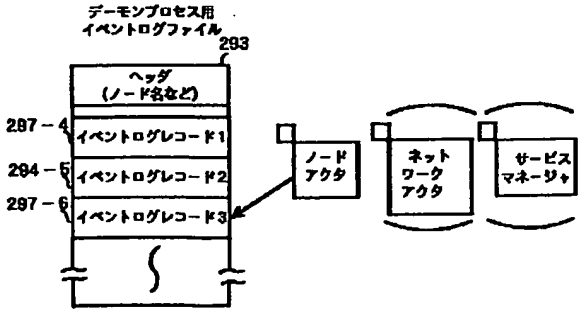
【図16】



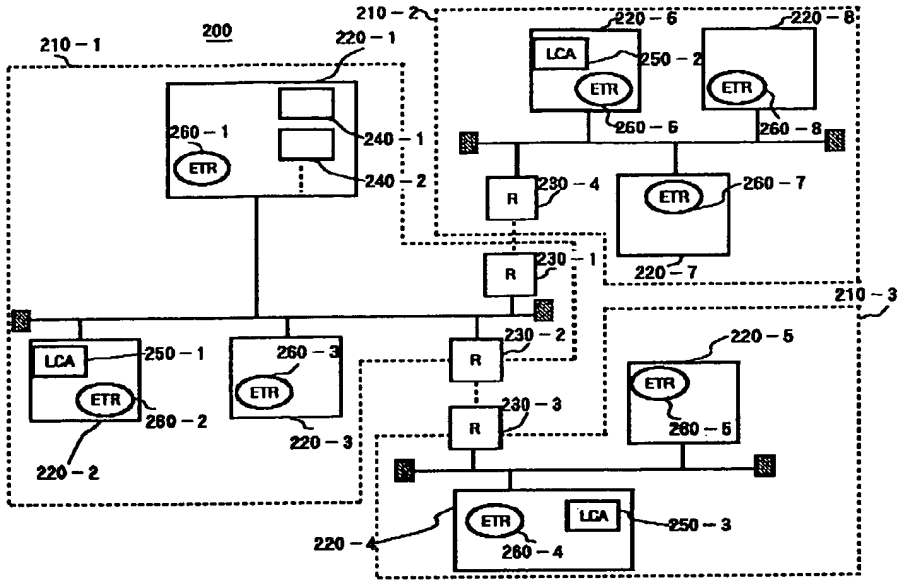
【図18】



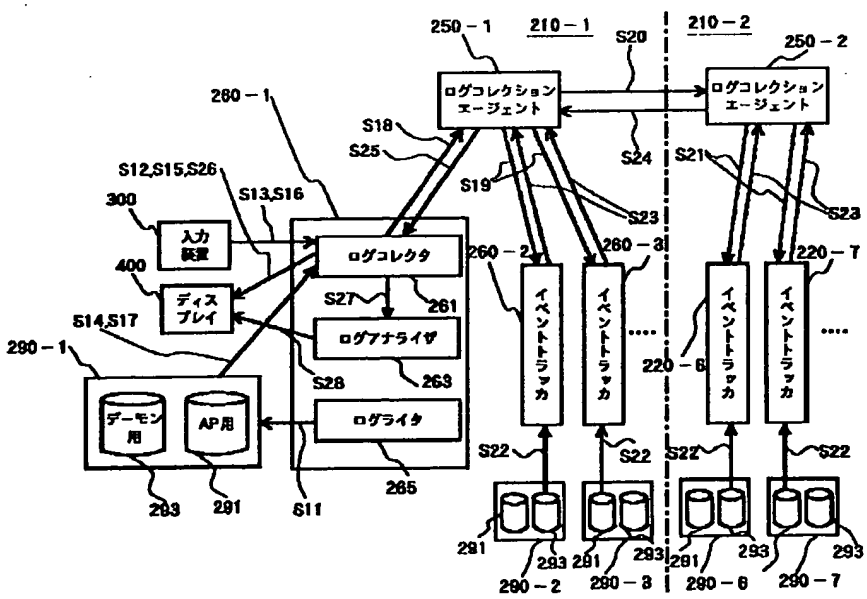
【図19】



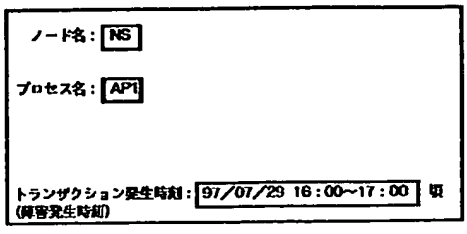
【図15】



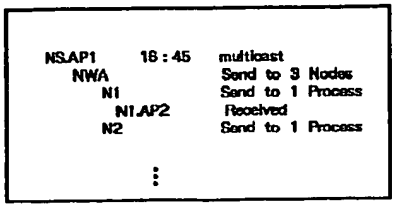
【図17】



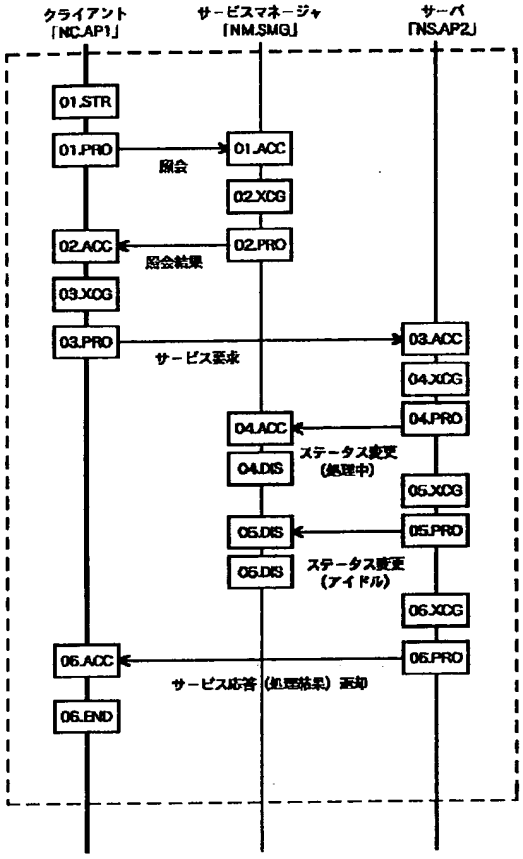
【図23】



【図26】



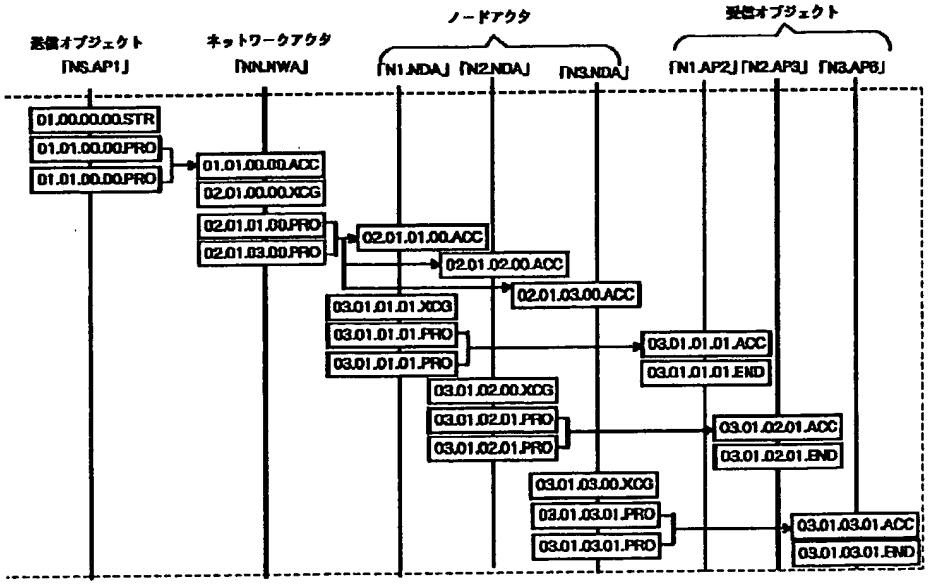
【図21】



【図25】

ID	オブジェクト	中継点カウンタ	種別	
579	NS.AP1	01.00.00.00	STR	97/07/29 16:45
579	NS.AP1	01.01.00.00	PRD	97/07/29 16:45
579	NS.AP1	01.01.00.00	PRD	97/07/29 16:45
579	NN.NWA	01.01.00.00	ACC	97/07/29 16:45
≀				
579	N1.NDA	02.01.01.00	ACC	97/07/29 16:45
579	N1.NDA	03.01.01.00	XCG	97/07/29 16:46
≀				
579	N3.AP6	03.01.03.01	ACC	97/07/29 16:46
579	N3.AP6	03.01.03.01	END	97/07/29 16:48

【図22】



フロントページの続き

(72)発明者　山本　英明
東京都江東区豊洲三丁目3番3号　エヌ・ティ・ティ・データ通信株式会社内
(72)発明者　増木　亮介
東京都江東区豊洲三丁目3番3号　エヌ・ティ・ティ・データ通信株式会社内
(72)発明者　内川　淳
東京都千代田区丸の内一丁目3番2号　株式会社住友銀行内
(72)発明者　市来　伸彦
東京都千代田区丸の内一丁目3番2号　株式会社住友銀行内